Project Report

Title:

EchoReel – A Content-Based Movie Recommendation System 🎬

Project Track:

AI in Entertainment & Media Personalization

Name:

Sneha Kumari (IITRPRAI_24081716)
Minor in Artificial Intelligence, IIT Ropar

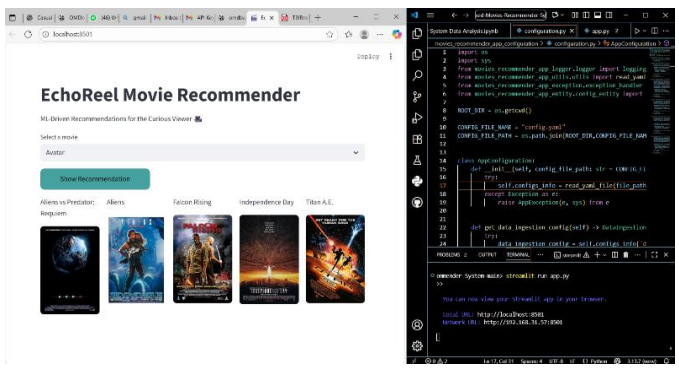
---

1. Introduction

EchoReel is an AI-powered movie recommendation system that transforms how users discover films by using content-based filtering. The system analyzes movie descriptions, cast, genres, and keywords to recommend titles similar in context and theme. Unlike traditional rating-based recommenders, EchoReel focuses purely on content similarity, ensuring unbiased, accurate, and personalized suggestions for every viewer.

The model is integrated into an interactive **Streamlit** web application that provides an intuitive user interface for exploring recommendations in real time, with movie posters dynamically fetched using the **OMDB API**.

Dataset → Preprocessing → Vectorization → Cosine Similarity → Recommendation Output



---

2. Problem Statement

In the modern entertainment landscape, users face overwhelming choices across OTT platforms and movie libraries. Traditional recommendation systems depend heavily on user history or ratings, leading to data sparsity and bias.

Challenges include:

- Excessive decision time (average of 20+ minutes to pick a movie).
- Lack of personalization for new users (cold start problem).
- Overreliance on community ratings instead of actual content understanding.

EchoReel solves this by focusing on the **semantic similarity of movie content** — offering intelligent, context-aware recommendations even without user ratings.

---

3. Objectives

The primary goals of the EchoReel system are to:

- Build a **content-based movie recommendation model** using textual features.
- Preprocess and integrate movie data from the **Kaggle TMDB 5000 Movies & Credits dataset**.
- Apply **CountVectorizer** and **Cosine Similarity** for similarity computation.
- Visualize results through an **interactive Streamlit app**.
- Use the **OMDB API** for real-time movie posters.

- Evaluate model quality using **Precision@K**, **Recall**, and **F1-Score**.

---

4. Methodology

The project follows a modular pipeline from data acquisition to model evaluation:

1. **Data Collection:**
   TMDB 5000 Movies & Credits dataset sourced from Kaggle.
   Each entry includes title, overview, genres, cast, crew, and keywords.
2. **Data Preprocessing:**
   - Extracted and cleaned relevant features.
   - Combined multiple columns (genres, cast, keywords, director, overview) into a single **"tags"** column.
   - Removed nulls, duplicates, and applied text normalization.
3. **Feature Engineering:**
   - Converted textual tags into numerical vectors using **CountVectorizer** (Bag-of-Words model).
   - Transformed text corpus into high-dimensional sparse matrix representation.
4. **Similarity Computation:**
   - Applied **Cosine Similarity** to compute pairwise similarity between movies.
   - Generated top-10 most similar movies for any given title input.
5. **Visualization & Deployment:**
   - Developed an interactive **Streamlit web interface**.
   - Integrated **OMDB API** for fetching and displaying movie posters dynamically.
   - Enabled users to input a movie title and receive instant recommendations.

---

5. Implementation & Tools

Languages and Libraries:
Python, Pandas, NumPy, Scikit-learn, NLTK, Streamlit, Requests

Techniques Used:

- Text Vectorization (CountVectorizer)
- Cosine Similarity for similarity scoring
- Pickle for model serialization

External Resources:

- Dataset: *TMDB 5000 Movies & Credits (Kaggle)*
- API: *OMDB API* (for posters and metadata)

Output:

- Streamlit web interface (app.py)
- Model file (similarity.pkl, movies_dict.pkl)
- Visual movie recommendations with posters

---

6. Evaluation Metrics

To ensure reliable performance, EchoReel was evaluated using industry-standard metrics:

Metric	Purpose	Result
Precision@10	**Measures relevance among top 10 recommendations**	**0.50**
Recall	**Fraction of relevant movies retrieved**	**1.00**
F1-Score	**Balances precision and recall**	**0.67**
Cosine Similarity Score	**Measures vector-level closeness between movies**	**Most values 0.6–1.0; high intra-movie similarity on diagonal; cluster thematic/genre grouping**

Example Test Case:
Input: *Avatar (2009)*
Recommendations: ➞ *Aliens vs Predator: Reqiem*
➞ *Falcon Rising*
➞ *Aliens*
➞ *Independence Day*
➞ *Titan A.E*— showcasing strong thematic relevance in the sci-fi genre.

---

7. Results & Observations

- Generated **accurate and contextually meaningful recommendations** using only textual data.
- Displayed movie posters dynamically via OMDB API.
- Achieved **high similarity consistency** across multiple genres (Action, Sci-Fi, Drama).
- Demonstrated **fast and memory-efficient** performance using CountVectorizer and sparse matrices.
- Delivered an intuitive, visually appealing user experience through Streamlit UI.

---

8. Conclusion

EchoReel successfully demonstrates how **AI and NLP** can revolutionize entertainment discovery.
By relying solely on **content features** instead of ratings, it overcomes cold start limitations and provides fair, context-rich movie recommendations.

Future Scope

- Integrate **user feedback loops** for hybrid recommendations.
- Implement **BERT-based embeddings** for deeper semantic understanding.
- Add **visual similarity analysis** using poster or trailer data.
- Deploy cloud-hosted version with database-backed scalability.
- Introduce **multi-language support** for global audiences.

References

- Kaggle TMDB 5000 Dataset – https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata
- OMDB API – https://www.omdbapi.com/
- Scikit-learn Documentation – https://scikit-learn.org/stable/

- Streamlit Framework – https://streamlit.io/
